WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Friday, November 19, 2004

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
□	L6	20001120	14
□	L5	L4 and ((record or recording or store or storing) near5 configuration)	28
□	L4	L3 and ((first adj3 configuration) near5 (second adj3 configuration))	35
□	L3	L2 and ((monitor or monitoring) near6 configuration)	1352
□	L2	L1 and (network near8 (intercommunication or interface))	23597
□	L1	(configure or configuration or configured) near8 (printer or device)	242482

END OF SEARCH HISTORY

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

□ Generate Collection

L6: Entry 1 of 14 File: USPT Nov 25, 2003

DOCUMENT-IDENTIFIER: US 6654797 B1
TITLE: Apparatus and a methods for server configuration using a removable storage device

Abstract Text (1):

Apparatus and methods for server configuration using a removable storage device are provided. The apparatus and methods include coupling a removable storage device reader to a server and inserting a removable storage device into the reader. The removable storage device includes configuration data that is used to configure the server. When power is supplied to the server, the server performs a boot-up sequence that includes uploading the configuration data from the removable storage device. The same removable storage device reader may be used to configure a plurality of servers. That is, the removable storage device reader is capable of being easily moved and coupled to a plurality of servers one after the other. The user may make use of the same removable storage device or different removable storage devices for each of the plurality of servers. The user is not required to have any technical knowledge regarding configuring server devices since the server performs the configuration itself upon boot-up based on the configuration information uploaded from the removable storage device.

Application Filing Date (1):
20000525

Brief Summary Text (7):
With known systems there are currently three methods by which a thin server can be configured. The first method requires the thin server to be configured using a keyboard and monitor directly attached to the thin server. This requires that the keyboard and monitor ports be included in the hardware, even though the intention is that once configured, direct access to the thin server will no longer be needed. The problem with this solution is that each device must be individually configured using a keyboard and monitor before being placed into production. This is a labor intensive activity and may involve additional handling of the thin server before it can be rack-mounted thereby increasing the possibility of accidental damage. This solution also increases the cost of the thin servers by requiring the additional hardware (keyboard and monitor) to be present even though the hardware is used only for configuration and is later not needed.

Brief Summary Text (8):
The second method of configuring a thin server involves performing the configuration using an LED panel and controls, e.g., push buttons, that are directly attached to the thin server. This requires that the hardware design include the additional cost of a front mounted input/output display console and the cost of embedding firmware to handle the configuration tasks. Another requirement is that the person has to be physically present at the device to enter configuration data. Again, the cost of the thin server is increased by requiring the additional hardware that is only used for configuration.

Brief Summary Text (12):
The present invention provides apparatus and methods for server configuration using

a removable storage device. The apparatus and methods include coupling a removable storage device reader to a server and inserting a removable storage device into the reader. The removable storage device includes configuration data that is used to configure the server. When power is supplied to the server, the server performs a boot-up sequence that includes uploading the configuration data from the removable storage device. In this way, the server is configured for use in a network.

Brief Summary Text (13):
In addition, the same removable storage device reader may be used to configure a plurality of servers. That is, the removable storage device reader is capable of being easily moved and coupled to a plurality of servers one after the other. The user may make use of the same removable storage device or different removable storage devices for each of the plurality of servers.

Brief Summary Text (14):
Furthermore, the user of the present invention is not required to have any technical knowledge regarding configuring server devices. Rather, the user merely need couple the removable storage device reader to the server, insert the removable storage device and supply power to the server. The server automatically uploads the configuration information from the removable storage device and configures itself using this configuration information.

Detailed Description Text (11):
A removable storage device reader 120 is shown in FIG. 1 coupled to the server 118. The removable storage device reader 120 reads configuration information from a removable storage device when the server 118 is first turned on. The removable storage device may be any type of storage medium upon which configuration data may be stored. For example, the removable storage device may be a floppy disk, a CD-ROM, a smart card, an optical disk, or the like. The removable storage device reader 120, therefore, may be any type of device capable of reading a removable storage device. For example, the removable storage device reader 120 may be a floppy disk drive, a CD-ROM drive, a smart card reader, an optical disk drive, or the like. In a preferred embodiment, for security purposes, the removable storage device is considered to be a smart card and the removable storage device reader 120 is a smart card reader, as will be described in more detail hereafter.

Detailed Description Text (12):
The server 118 is provided with boot instructions such that, upon power-up, the server 118 sends a request to the removable storage device reader 120 to read configuration data from the removable storage device inserted therein. This configuration data may include, for example, the IP address of the server, the hostname, the netmask, the gateway, domain and nameserver information for the server 118.

Detailed Description Text (13):
The configuration data is read from the removable storage device by the removable storage device reader 120 and provided to the server 118 via a wired or wireless connection between the removable storage device reader 120 and the server 118. In this way, the server 118 is configured using the configuration data read from the removable storage device.

Detailed Description Text (17):
Additional PCI bus bridges 222 and 224 provide interfaces for additional PCI buses 226 and 228, from which additional modems or network adapters may be supported. In this manner, server 200 allows connections to multiple network computers. A memory mapped graphics adapter 230 and hard disk 232 may be connected to I/O bus 212 as depicted, either directly or indirectly.

Detailed Description Text (22):
With the present invention, the same removable storage device reader may be used to

configure a plurality of servers. That is, the removable storage device reader is capable of being easily moved and coupled to a plurality of servers one after the other. The user may make use of the same removable storage device or different removable storage devices for each of the plurality of servers.

Detailed Description Text (23):
Furthermore, the user of the present invention is not required to have any technical knowledge regarding configuring server devices. Rather, the user merely need couple the removable storage device reader to the server, insert the removable storage device and supply power to the server. The server automatically uploads the configuration information from the removable storage device and configures itself using this configuration information.

Detailed Description Text (27):
The smart card in accordance with a preferred embodiment of the present invention stores configuration data for one or more servers. The smart card can be configured with the configuration data prior to or subsequent to the physical setup of the server. Thus, the physical setup and the configuration of the device can be allocated to different personnel with different levels of technical expertise, security clearances, etc.

Detailed Description Text (30):
With the present invention, instead of looking for an operating system kernel, or in addition to looking for an operating system kernel, the boot code of the server instructs the server to look for a configuration profile in a predetermined location in local memory/storage. If a configuration profile exists, then the server is already configured and need not be configured using the removable storage device reader.

Detailed Description Text (32):
If a configuration profile does not exist, the server looks to a specific location for a configuration profile on a smart card. This specific location is based on a determination made by the server of which devices are coupled to the server. For example, the server may have settings indicating an order in which devices are to be searched for configuration profiles. This may be similar to setting the boot sequence in BIOS with conventional computers.

Detailed Description Text (33):
In response to finding a configuration profile on the removable medium, in this example a smart card, the smart card reader reads the configuration profile data from the smart card and stores it in the server. The server is thereby configured for use in the appropriate network.

Detailed Description Text (37):
A determination is made as to whether a configuration profile is found (step 404). If not, the server retrieves a configuration profile from a smart card location and stores it in local storage (step 405). If so, the server is already configured and no further action to configure the server is necessary. Thus, the operation ends.

Detailed Description Text (39):
Thus, the present invention provides a mechanism by which servers may be configured without required extensive extra hardware to be incorporated into the servers. The mechanism allows a user to quickly configure one or more servers using a removable storage device reader. In this way, the removable storage device can be created at a remote time and/or place from the creation of the installation, configuration, and setup of the server itself. Furthermore, the user is not required to manually input configuration information for each server at the time of installation and setup. Rather, the removable storage device stores the configuration profile that is uploaded to the server when the server is powered on and executes a boot-up sequence.

CLAIMS:

13. The method of claim 9, wherein the first server configuration information and second server configuration information include one or more of an IP address of the server, a hostname, a netmask, a gateway, a domain or a nameserver.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

□ Generate Collection

L6: Entry 4 of 14 File: USPT Sep 5, 2000

DOCUMENT-IDENTIFIER: US 6115713 A
TITLE: Networked facilities management system

Application Filing Date (1):
19960830

Brief Summary Text (23):
In some cases, it is desirable to obtain reports by routing messages to devices which were not part of the network when configured. For example, ease of maintenance may be enhanced by allowing connection of a personal computer (PC) to an unoccupied port on a network node. It may also be desirable to provide other non-configured devices, such as printers, access to the nodes on the network. Traditional systems restrict the use of such non-configured devices, since there is no way to communicate with a device whose presence has not previously been made known to the network, for example, by assignment and storage of an address.

Brief Summary Text (24):
As previously noted, networked systems have at least 2 nodes with components for performing processing functions appropriate to the system and communicating with each other over communication links. In a facilities management system (FMS) such nodes can contain processors, A/D and D/A converters and other equipment interface circuits to obtain sensor data required for processes implemented in the node and to issue equipment commands. The communication links include various communication media facilitating communication among nodes on the same bus, subnet or network or between nodes on different networks over gateways. Nodes are configured on a system when they are defined in one or more storage devices as members of a network. Node configuration may occur by storing data defining a path to the node. Thus, the system has knowledge of the node's existence. Depending on the system, storage of configuration information may be centralized or distributed. Such configuration information may include data indicating the type of node, its location on the system, and other information defining a path to the node.

Brief Summary Text (27):
Dynamic or adaptive routing strategies route messages over communications links in response to message traffic and topology. Adaptive strategies include centralized, isolated or decentralized, and dynamic routing. Centralized routing strategies have a central node monitoring the number and length of messages transmitted over communications links and dynamically issuing routing strategies based on message traffic patterns. This is usually accomplished by updating and changing routing tables in response to the changing traffic patterns. Decentralized strategies distribute partial routing tables among the nodes. For example, when a message is routed to an intermediate node along a path to its final destination, the intermediate node examines the traffic pattern among alternative remaining paths to the destination node and dynamically selects one of the several alternatives according to certain measures of efficiency. Thus, adaptive strategies provide for reconfiguring routing tables in response to changed conditions, including the addition of new devices. However, in many cases it is not possible to incorporate non-configured devices. Even where this is possible, the temporary incorporation of a previously non-configured device often does not justify the added processing

required for dynamically adjusting routing tables. Such processing increases message transmission time and reduces overall system efficiency.

Brief Summary Text (28):
Regardless of the routing strategy employed by various parts of the system, in certain applications, such as maintenance, diagnostics, and administrative functions, it is desirable to allow data communications between a node on one of the communications links in the system and a temporary node or processing device. This is particularly true in automated networked control systems. Such systems often have need for emergency maintenance and diagnostic activities and for temporary load analysis. Present techniques are cumbersome because these require temporarily disabling at least portions of the network while a new node is configured onto the network. Configuring new nodes on a network is difficult since new data communication path strategies must be worked out. Moreover, developing temporary data path strategies could result in inefficient communication strategies between the temporary or non-configured device and the nodes configured on the network.

Brief Summary Text (86):
allowing devices not included in the original network configuration to communicate with configured nodes on the network.

Brief Summary Text (87):
It is another object of the invention to allow such non-configured devices to receive messages from configured nodes on the network.

Brief Summary Text (88):
It is yet another object of the invention to allow such non-configured devices access to networks which employ either adaptive or non-adaptive routing strategies.

Brief Summary Text (89):
It is a further object of the invention to allow such non-configured devices access to the network without requiring the down loading or updating of static or dynamic routing tables in existing nodes.

Brief Summary Text (90):
It is a further object of the invention to allow such non-configured devices to be attached to a first configured node on a network using one of either an adaptive or non-adaptive routing strategy and to receive messages from other nodes on other networks using the same or a different routing strategy.

Brief Summary Text (91):
It is a still further object of the invention to allow such non-configured devices access to a network without requiring shutdown of the system.

Brief Summary Text (215):
The above objects of the invention are further accomplished by attaching a non-configured device to a port on a configured node of a network. The non-configured device, which contains its own process identifiers, communicates via that port with the configured network node. The configured network node communicates with other configured network nodes to route messages from the non-configured device to their destinations. Destination nodes recognize the message source as the configured network node or as a non-configured device dropped from a port on a configured node. Thus, at the destination node, responses generated are transparent to the status of the source as a non-configured device. The final destination node responds as though the message is from a configured node and the response message follows the same or an alternate data communication path back to the configured node having the non-configured device connected to its port. Based on communications over a drop between the non-configured node and the configured node,

the configured node provides the message to the non-configured device which delivers it to a process identified in the message. This allows any configured node to respond to data requests made by a non-configured device.

Drawing Description Text (39):
FIG. 37 shows an example of a non-configured device attached to a configured node for communicating over communications links with one or more configured nodes.

Drawing Description Text (41):
FIG. 39 illustrates the transmission of a request from a non-configured device or a response from a configured device.

Drawing Description Text (42):
FIG. 40 illustrates the receipt of a request from a non-configured device or the receipt of a response from a configured device.

Drawing Description Text (43):
FIG. 41 tabulates a possible routing strategy for messages between the non-configured device and a configured node.

Detailed Description Text (2):
FIG. 1 shows generally network control module 1-1 which has a processor 1-3, dynamic random access memory 1-5, and electronically programmable read only memory 1-7. Network control module 1-1 communicates with high speed bus 1-9, the N1 bus, so that network control module 1-1 can be interconnected in a local area network configuration to other network control modules. A plurality of network control modules 1-1 connected over high speed bus 1-9 form a network which can be interconnected through gateways to other networks of network control modules interconnected on high speed buses. Network control module 1-1 further has standard RS-232 interface 1-11 with a plurality of ports to provide communication through a modem over port 1-13, a specialized network terminal over port 1-15 and a computer, or printer over port 1-17. Field trunk controllers 1-19 and 1-21 allow network control module 1-1 to communicate with field devices interconnected on communications media 1-23 and 1-25.

Detailed Description Text (8):
As shown in FIG. 3, expansion module (XM) 3-1 according to the invention, includes processor 3-3 and memory 3-5. The memory is typically divided into static random access memory (SRAM) 3-7 and electronically erasable programmable read only memory (EEPROM) 3-9. Point multiplex modules 3-11 provide a configurable input/output for the expansion modules. The expansion module is also a plug in module which plugs into a connector on a back plane in a node of a facilities management system. The expansion modules condition binary, analog and pulse inputs and report changes to the network controller or network control module 1-1. In addition, the expansion module executes binary output commands from network controller 1-1. Point multiplex modules 3-11 provide five configurations of expansion modules. These include a first configuration having 32 binary inputs, a second configuration having 8 binary inputs and 8 pairs of outputs using momentary relays, a third configuration having 8 binary inputs and 8 magnetically latched relay outputs, a fourth configuration having 8 analog inputs and a fifth configuration having 8 binary inputs and 8 electrically maintained relay outputs.

Detailed Description Text (11):
As shown in FIG. 1, network control unit 4-1 is provided an interface to an operator through RS-232 interface portion 1-11 of network control module 1-1. Using a dial-up modem 1-13, specialized network terminal 1-15 or an operator work station such as a personal computer, an operator may generate or respond to commands and provide program changes and organizing data to user specified data bases.

Detailed Description Text (17):

As previously stated, a network control unit must have a network control module in order to accomplish peer-to-peer communications over the N1 bus. However, as the single slot configuration in FIG. 6 shows, it is possible for a device to be constructed having an expansion module without a network control module. Since an expansion module could not communicate over the N1 bus, the device can not be a network control unit. It is possible, according to the invention, to construct in either the 5 slot back plane shown in FIG. 4, the two slot back plane shown in FIG. 5 and the one slot back plane shown in FIG. 6 a device which does not have the capability of communicating over the N1 bus. Such devices are called network expansion units (NEU). Network expansion units serve two functions. First, they serve as a collection platform for I/O points in order to increase the point and control loop capacity of an NCU. Second, network expansion units can be remotely located from an NCU to monitor and distribute control to points and then transfer the data from these points back to the NCU over the N2 bus.

Detailed Description Text (19):
FIG. 9 illustrates a possible configuration of a facilities management system according to the invention. Five slot NCU 9-1 communicates with one slot NCU 9-3 over N1 bus 9-5. N1 bus 9-5 is also connected to personal computer 3-7. Personal computer 9-7 can be used as a download device to download new information and data bases to NCUs 9-1 and 9-3 and other devices connected to NCUs 9-1 and 9-3. N1 bus 9-5 is connected to communication terminal board 9-9 in NCU 9-1 and terminal communication board 9-11 in NCU 9-3. Within NCU 9-1 N1 bus 9-13 is connected to network control module 9-15. Since this is the only network control module shown in five slot NCU 9-1, there are no further connections within the five slot NCU to the N1 bus 9-13. Five slot NCU 9-1 also has expansion modules 9-17 and 9-19 and digital control modules 9-21 and 9-23. These modules perform the functions discussed previously and are interconnected with the five slot NCU via N2 bus 9-25. An interface communicates directly with the five slot NCU 9-1, N1 bus 9-13 and device on its N2 bus via lap-top PC 9-27. As FIG. 9 shows, lap-top 9-27 is connected to an RS-232 interface 9-29 which is part of network control module 9-15. A network terminal unit for specialized network terminal 9-31 is also accommodated on RJ-11 interface 9-33. Network control module 9-15 also has sub-modules 9-35 and 9-37. Such sub-modules may include devices such as subnet controller 1-27, field truck controller 1-21 and RS-232 interface 1-11. Function modules, for example, 9-41 are also used in the five slot NCU 9-1. As FIG. 9 shows, each device in the five slots has its own power supply, for example, 9-43. The power supplies all receive line voltage or AC power through power terminal board 9-45. The individual power supplies exist to isolate spurious signals and noise in one device from being communicated via the power supply lines into a second device.

Detailed Description Text (35):
Software features at the highest level communicate with the software object level 17-7. The software object level is an intermediate level which determines how to carry out the action requested by one of features 18-21, 18-23, 18-25 at the features level 18-5. Information is passed between the software object level 17-7 and 18-7 and the features level 17-5 and 18-5 independent of differences at the lower levels of hardware. Similarly, the software object level forms an interface with another level of software called the hardware object level 17-9 and 18-9. The hardware object level allows for a common interface to be established between the software object and hardware object levels regardless of the peculiarities of operational units, such as sensors and other data acquisition and control instruments, connected to the network controller. This is accomplished by configuring the hardware object level to mask the differences between the operational units to the various entities in the software object level 17-7 and 18-7. In accordance with requirements of local bus communications interface 17-11 and 18-11, network controller 17-1 communicates over local bus 17-13 with slave controllers 17-15, 17-17, and 17-19. As shown in FIG. 1, any number of types of slave controllers is possible. The slave controllers are connected to operational units, for example, to sensors. Such sensors are binary or analog field sensors

which read values of real world data (e.g., outdoor air temperature).

Detailed Description Text (49):
It should be further noted that as shown in FIG. 18, communication is possible between all the features and all the object managers in software object level 18-7 and all the object managers in software object level 18-7 and hardware object level 18-9. The actual communications paths used are a function only of the function performed by the feature and the data required. Thus, Feature y may also request software object T1, thus accessing analog input object manager 18-27. Similarly, Feature n may request data from one or more object managers in software object level 18-7 which may further request data from one or more object managers in hardware object level 18-9. The commonality of interface between the hardware object and software object level simplifies the addition of new slave controllers and object instances. An object instance would be added in the above example if a fourth temperature sensor T4 were to be added to the system. A new slave controller of the same type would be added if a third type A slave controller, A3, were added. In both cases, all the necessary software exists on the network controller because there are no changes to the informational interfaces between the software object level 18-7 and the hardware object level 18-9. The user need only modify the database to create a new instance of the 18-29 analog input object T4 or the database 18-39 to create another instance of type A controller object, e.g. A3, in the network controller.

Detailed Description Text (87):
controlled by the node. In function block 28-5, a node is configured on a network by giving it an N1 address and storing the identity of its archive device in non-volatile memory. Following power-up at step 28-7, the node must be synchronized with the other nodes at step 28-9. Step 28-11 tests if the synchronization is complete. If not, control is transferred back to function block 28-9 to complete the synchronization process. Upon completion of node synchronization, control transfers to function block 28-13 in which the node accesses the archive device to download its own particular data base. As each data element is received, decision block 28-15 tests if the information is to be transmitted to a device on the N2 bus. If not, the information is stored in the node as shown in function block 28-17. If the information is destined for a device on the N2 network, as shown in function block 28-19 the information is passed onto the device through the N2 network.

Detailed Description Text (139):
According to a routing convention in FIG. 37, each node is identified by a network address. The elements of a network address include at least three fields: first, an identifier of the communication link called a subnet, and second, a local address of the node on the communication link or subnet. For example, node 37-9 is at subnet 2 local address 1. The third field of the network address is the port number of the node from which a device is dropped, called the Drop ID. As illustrated in FIG. 37, each individual configured node itself is Drop ID 0. Non-configured devices, such as lap-top computers or other data processing devices, can be connected or dropped to numbered ports of the node. Here it is again understood that the present invention accommodates any number of node ports and introduces no limit on such node port capabilities. A port of non-configured lap-top computer 37-13 can be connected to a port from a node, such as node 37-3 and assigned a network address. For example, if port 2 of non-configured lap-top computer 37-13 is connected to port 3 (Drop ID 3) of node 3 which is at subnet 1, local address 1, the network address of lap-top 37-13 is 1:1:3 as shown in FIG. 37. It should be noted that the port of lap-top computer 37-13 is not part of the network address. FIG. 37 further illustrates that another lap-top computer 37-15 can be part of the network as originally configured. According to the naming convention, such devices are identified as additional nodes on additional subnets, in this case, subnet 3, local address 1, Drop ID 0.

Detailed Description Text (141):
According to the invention, when a non-configured device is attached to a port of a configured node, the non-configured device establishes its presence on the port. When receiving messages from other configured nodes the configured node first determines from the subnet and local address destination portions of the message if it is the destination node. If not, the message is passed on to the next proper configured node defined by the route. At the destination, the configured node evaluates the Drop ID of the received messages to determine if the message is for itself (Drop ID 0) or for the attached non-configured device (non-zero Drop ID).

Detailed Description Text (142):
FIG. 39 illustrates the generation and transmission of a message by a process on non-configured lap-top 37-13 which seeks to communicate over the network with another device. To initiate the communication request shown in block 39-301 an initialization phase first takes place in which the non-configured device establishes its location on the network. Non-configured device 37-13 sends a message requesting the address of the node or FMS network controller to which it is attached, in this case node 3. The node or FMS network controller responds by activating an initialization task which sends the network address including the subnet, local address and Drop ID back to the non-configured device. The non-configured device then stores this information as its network address.

Detailed Description Text (143):
In function block 39-303 the non-configured device accesses this address and uses it as the source address portion of messages it generates. These messages include both the source address and destination address and data or data requests to be transmitted. For illustration, assume that non-configured lap-top 37-13 has requested data concerning the status of a damper 37-16 recorded in configured lap-top 37-15. In function block 39-305 the processor in the node transmitting the message determines if the request is for a process remotely located in another node or for a local process in this node. If not, as shown in function block 39-307, the request is delivered to the local process and exit 39-309 is taken. If the request is for a process in another node, function block 39-311 determines if the source and destination network addresses are valid. This requires that network processing layer 38-201 in the node verify that the subnet, the local address of the node or network controller on the subnet, the Drop ID and the process identifier are valid. If not, error processing 39-313 begins and exit 39-309 is taken. If the network addresses are valid, the network layer 38-201 in the first node references a routing table stored in a memory 37-6 to determine the next hop in the path. As previously discussed, such routing tables may be static or dynamic, centralized or decentralized. For illustrative purposes only and not as a limitation of the invention, a static routing table is assumed. The request is then tagged with the network address of the transmitting node for acknowledgement by the next intermediate destination in the data link layer 38-203 of the node, as shown in function block 39-317. Transmission of the request then takes place in function block 39-319.

Detailed Description Text (144):
As discussed above, FIG. 39 illustrates the activities involved following a request from a non-configured device to communicate over the network. The same processing takes place when a node or network controller transmits a response from a configured device. Thus, by using the same processing that takes place when a network controller or node transmits a response from a

Detailed Description Text (145):
configured device, a request by a non-configured device to communicate over the network can be accommodated.

Detailed Description Text (147):
FIG. 40 illustrates the activities of any given node which take place upon receipt

of a request from a non-configured device. These activities are the same as those that take place upon the receipt of a response from a configured device. Thus, the same approach for handling receipt of responses from configured devices can be used to respond to a request from a non-configured device. As previously discussed, messages from configured nodes are tagged by the forwarding node so that receipt can be acknowledged. As shown in FIG. 40, in function block 40-403 the message is first evaluated to determine if the tagged message is from a valid source to a valid destination and whether the message is appropriately tagged, as previously discussed relative to FIG. 39. If not, as shown in function block 40-405, the message is discarded and an exit 40-407 is taken. In addition, other known tag functions for reliability such as sliding windows, can be performed. If the processing in the data link layer 38-203 in function block 40-403 identifies the message as valid, function block 40-409, also a part of the data link layer 38-203, transmits an acknowledgement of receipt of the message to the forwarding node. At the network layer 38-201, the message is tested in function block 40-411 to determine if the destination process is located at the receiving configured node. If so, function block 40-413 delivers the request to a process local to the receiving node and takes exit 40-407. If the destination process is not located at this node, network layer 38-201 processing continues as shown in block 40-415. The destination process is then tested to determine if it is for a non-configured node. If this is the case, the network layer readdresses the response for a non-configured device, the data link layer retags the response and it is then transmitted, as shown in function blocks 40-417, 40-419 and 40-421 respectively. If the processing in block 40-415 is such that the destination process is not at a non-configuring node, the request is readdressed for the next hop in the path, retagged, and transmitted as shown in blocks 40-423, 40-425, and 40-427 respectively. It should be noted that processing in function blocks 40-409, 40-427 is the same for receipt by any node in the system.

Detailed Description Text (148):
FIG. 41 shows that the response from configured lap-top 37-15 can be routed to non-configured lap-top 37-13 at network address 1:1:3 by retracing the exact message path previously traversed in going from non-configured lap-top 37-13 to configured lap-top 37-15. Using this approach, it is not necessary to actively evaluate an additional data communication path to return the information required by the non-configured device. The response from lap-top 37-15 containing the status of damper 37-16 is routed back through the nodes to node 37-3 which, as previously discussed, tests the messages it receives to determine if the message is destined for the node itself or for the non-configured device on the node. In this case lap-top 37-15 addresses the response to the source of the request, identified as network address 1:1:3. Since node 37-3 at subnet 1, local address 1 recognizes the Drop ID 3 as the node port attached to non-configured device 37-13, the response is sent to lap-top 37-13.

Detailed Description Text (149):
Finally, it should be noted that a response from configured node 37-15 to the request from non-configured device 37-13 need not traverse the same path. For example, in adaptive routing systems variations in message traffic conditions may result in the response traversing a different path through the network than the request. Indeed, it is possible for the network on communications link 37-4 to employ a static routing scheme, while the network on communications link 37-17 employs an adaptive routing, or vice versa. All adaptive, all non-adaptive, or any combination of networks can be used with the invention. However, regardless of how the response reaches the configured node, the configured node routes the message to the non-configured device based on the Drop ID in the network address given as the destination of the message. As a result, functions which are not normally incorporated into the network can be performed by attaching a non-configured device to a convenient port from one of the nodes on one of the networks in the system. This is because the Drop ID of the network address allows responses from configured nodes to be routed to non-configured devices dropped from ports on configured

nodes.

Detailed Description Text (218):
By way of example, in facilities management system 59-8, a high level software feature programmed to perform control system functions requests data transfers between itself and an analog output object which is one of the software objects 59-12 programmed into a memory 59-16 of a network controller 59-2. Network controllers 59-2, 59-4 each may function to control activities of digital control modules 59-18 and 59-20. The analog output object can map to hardware objects 59-14 and the network controllers 59-2 and 59-4 interface with digital control modules 59-18 and 59-20. Software objects 59-12 are stored in a database and are controlled by a database manager for objects of that type. As discussed below, analog output objects can be analog output digital objects or analog output setpoint objects. Thus, an analog output digital (AOD) object 58-114 is manipulated by analog output object manager 58-100. Objects, including the analog output digital object 58-114, are structured into methods and attributes as described previously.

Detailed Description Text (265):
FIG. 69 shows a control loop with PID processing. The PID data base manager first provides an interface to other tasks in the network which may read data, e.g. 69-5, from a PID loop, write to a PID loop, or command a PID loop. The second PID data base manager task is to schedule processing of each of the 16 instances of PID loops. The third responsibility of the PID data base manager is to execute an auxiliary signal switch processing, output filter processing, high/low signal select processing and reliability switch processing in accordance with the inputs to these processing functions.

Detailed Description Text (278):
The PID database manager also accommodates two types of reads record message and causes the current CONFIGURATION read record message causes the current working definition for the given PID loop to be formatted and sent back through the N2 bus to the network controller. The other read record message is READ CURRENT STATE. This causes information on the current state of the PID loop along with values used during the last iteration of the processing to be sent via the N2 communication bus to the network controller.

Detailed Description Text (386):
The fault tolerant controller block executes once every twenty sampling periods of the PID controller. The process monitor and output switch functions execute once each sampling interval of the PID controller. In one system configuration shown in FIG. 72, the functions of the process monitor 72-1 and out put switch 72-5 can be implemented directly in a Digital Control Module 72-5 while the fault tolerant controller functions are implemented in the Network Controller 78-7.

CLAIMS:

1. A method of allocating high reliability data in a system including nodes communicating with each other over a network, the nodes each including a processing means for executing, a sensor interface for receiving sensor data elements from sensors in the system, and a memory means for storing programmed high level features, the processor means executing the programmed high level features to control a process, the method comprising steps of:

storing in the memory means of a first node of the nodes expected ranges of values of at least one of the sensor data elements for use by the high level features;

receiving a received data element of the sensor data elements from the sensor interface of the first node at the programmed high level features and comparing the received data element with an expected range of the ranges of values corresponding to the received data element in the first node at the programmed high level features;

if the received data element is within the expected range, tagging the received data element with an indicator that the received data element is reliable and using the received data element in further execution of the programmed high level features to control the process in the first node;

if the received data element is not within the expected range, seeking via the network from a second node of the nodes the received data element and substituting a value representative of the received data element within the expected range obtained from the second node; and wherein the step of seeking from the second node further comprises searching directories in the second node of the nodes for alternate storage locations for the received data element and for an alternate sensor of the sensors for producing a sensor data element corresponding to the received data element.

Previous Doc Next Doc Go to Doc#